

# Correction

NSI - 2021 Étranger Jour 1 (21-NSIJ1G11)

## Exercice 1 - Cryptage selon le « Code de César »

1. Ce code affiche dans la console :

D  
A

2.

```
def cryptage(self, texte):
    """ Retourne le texte chiffré """
    code = ""
    for i in texte:
        code += self.decale(i)
    return code
```

**Remarque :** On parle plutôt de « chiffrement » que de « cryptage ». Seule la notion de décryptage a un sens puisque qu'elle consiste à décrypter un code sans en connaître la clé. On parle de « chiffrement » puisque la clé est forcément connue au moment du chiffrement, on ne peut pas crypter sans connaître la clé de chiffrement.

3.

### Solution 1

```
# Demande de saisir la clé de chiffrement
cle = int(input("clé de chiffrement :"))
# Crée un objet de classe CodeCesar
cesar = CodeCesar(cle)
# Demande le texte à chiffrer
texte = input("texte à chiffrer :")
# Affiche le texte chiffré
print(cesar.cryptage(texte))
```

### Solution 2 (Plus robuste)

```
try:
    # Demande de saisir la clé de chiffrement
    cle = int(input("clé de chiffrement :"))
    if -26 < cle < 26:
        # Crée un objet de classe CodeCesar
        cesar = CodeCesar(cle)
        # Demande le texte à chiffrer
        texte = input("texte à chiffrer :")
        # Affiche le texte chiffré
        print(cesar.cryptage(texte))
except ValueError:
    print("La clé est incorrecte.")
except KeyboardInterrupt:
    pass
```

4. Il s'affiche dans le console :

FIN

Cette ligne de code crée une instance de la classe `CodeCesar` avec comme attribut une clé de chiffrement de **10** puis appelle sa méthode `transforme` avec comme argument la chaîne de caractère "**PSX**".

Cette méthode transforme la clé de chiffrement en son opposé et appelle la méthode `cryptage`. Ainsi, au lieu de décaler de 10 places vers la droite, le chiffrement du texte, décale de 10 places vers la gauche :

P devient F

S devient I

X devient N